

本周工作（2012.10.15-2012.10.21）

马晓红

周一把夏菁师姐要报告的那篇关于 **storyline** 论文的看了一下。就像陈老师在讨论的时候说的，这篇文章是把一个通常用手工来完成的工作通过计算机程序的方式来实现，我们之前在本科的时候也做过通过计算机编程绘制手绘地图等方面的工作。

本周继续看了一些关于 **information retrieval** 的论文，把之前看的与 **uiuc** 的工作内容比较接近的论文和本周写了个总结，以加深对这些论文的认识。因为是刚刚入门读这些文章，所以一遍看下来可能认识不会很深刻，有的也可能不能很好地理解论文的主要思想，通过写总结的方式可能会好一些。

1. **Web Object Retrieval**: 考虑到网络上信息的不准确，希望构建一个对数据的错误不敏感的效果稳定的检索模型。对网络上信息的两种表示方法 **record-level representation** 和 **attribute-level representation**（信息更多，误差也更多）各有优缺点，分别采用这两种表示方法和它们的综合工三种方法来构建语言模型，结果表明两者的结合更好。

我觉得采用的语言模型就是一个简单的语言模型，对于构建的三个模型表达式比较简单也比较容易理解。在 **evaluation** 中，为了说明数据误差对模型的影响采用了多个数据来源和多种信息提取方法。我自己觉得这篇文章可以借鉴的地方是考虑了信息提取中的误差，反映在模型上就是对不同层次的提取赋予不同的准确度值和误差的传播值。

2. **Language-model-based Ranking for Queries on RDF-Graphs**: 对 **RDF** 图上的结构化查询，采用了一种适用性比较广泛的统计语言模型，既支持非关键字的查询也支持关键字的查询，并对查询的结果进行了排序。当查询的结果比较少的时候，对查询的条件也可进行适当的放松以获得更多的结果。语言模型和排序模型是一种基于统计的模型，在介绍模型的时候论文里采用实例对其进行了说明，比较容易理解。在语言模型中把一个 **triple** 看成一个整体。对结果的排序根据结果的语言分布模型与查询问题的语言分布模型之间的 **KL-divergence** 来计算的。进行实验评估时，对比了论文中的方法与 **WOR**、**BANKS**、**NAGA** 这三种方法，结果表明论文中的方法更好。根据分析：**WOR** 只支持对 **entity** 进行排序，没有考虑 **entity** 之间的关系，所以作者认为当目标是一个排序图时，最好把 **triple** 看成一个整体；**BANKS** 的排序方法是依赖于结果图中的一些静态属性（节点和边的权重），所以作者认为可能不能很好地表示结果的质量；**NAGA** 是与论文中的方法比较接近的一种，性能也是最接近的（略差），但是这种方法既不支持近似匹配也不支持关键字扩展，在纯粹的结构化查询中结果与文章中的结果很接近，当测试文章模型中对查询条件的放松时，**NAGA** 就不能得出有效的结果了。

我觉得这篇文章的对查询条件的一个放松是比较好的方向，但是这篇文章是针对在结构性数据库上的结构化查询，对结构化的查询有一个很明确的放松方法，不知道对于非结构化的文本查询条件放松会不会不太容易。

3. **Exploiting Web Search Engines to Search Structured Databases**: 这篇文章的整体思想是要在网络搜索结果和结构型数据库之间建立一种联系，使得检索的结果能够定位到数据库中的某个 **entity**。整体的构架是先从网络上爬一些 **Document**，在 **Document** 中进行 **entity extraction** 然后是 **entity retrieval** 和 **entity ranking**。为了使用更少的时间和空间以及达到更好的效果，在具体的实现中每一步都考虑了很多具体的细节实现问题。在 **entity extraction** 中，传统的方法是对 **document** 进行分词、过滤得到相应的 **entity class**，在该论文中只提取数据库中存在的 **entity**，所以采用一种纯粹的语法匹配的方法，并且构建 **trie** 数来减少匹配的次数。考虑到具体网页上 **entity** 的名字可能会有缩写等方面的问题，

还通过构建一个 **reference table** 进行近似匹配。为了达到更高的准确率，对于 **document** 中的每一次 **entity mention** 判断其实 **true mention** 和 **false mention**（相同的 **entity name** 可能代表不同的含义等原因），采用 **SVM** 进行 **true mention** 和 **false mention** 的分类，利用 **wikipedia** 建立训练集，并考虑了其他的一些 **feature**（包括 **document-level**、**entity context**、**entity frequency**、**entity length** 等）。进行 **entity ranking** 时，也是考虑了多种影响的因素，包括 **entity frequency**、**proximity**、**document importance**。在 **experimental evaluation** 中，比较了这种结合了 **web search** 和 **structured database** 的结果和只考虑 **structured database** 的结果，在三个数据库上测试了 **entity ranking** 的方法，以及这种方法和效率和准确率。总的来说，在网络搜索结果和结构型数据库之间建立联系这种方法很大程度地提高了在结构型数据库上检索的效率。

我觉得这篇文章一个很大的特点是为了使用更少的时间和空间以及达到更好的效果，在具体的实现中每一步都考虑了很多具体的细节实现问题，在各个方面也都尽量全面地进行考虑。当然这其中很多也是值得借鉴的，比如在 **entity extraction** 时只考虑 **database** 中存在的 **entity** 以便很大程度地提高效率；为了提高准确率在 **entity extraction** 后进一步判断每一次 **entity mention** 是否是对的；为了提高效率能离线进行的就尽量离线进行。在这篇文章中有一点不懂的是在进行 **entity mention classification** 时利用 **wikipedia** 构建训练集，对于每一个 **entity string** 如果能够 **link** 到该 **entity** 就认为是 **true mention**。我觉得这好像依然不能解决 **false mention** 这个问题，因为好像对于每一个确定的 **entity string**，它在 **wikipedia** 中的 **link** 的一定的吧。这是个疑问。另外这篇文章的整体思想与目前的 **uiuc** 的实现思想有一部分是一致的，有很强的借鉴性。

4. **A Language Modeling Approach to Information Retrieval**: 这篇文章很老了（1998），里面介绍的也是概率语言模型，是一种非参数化的、整合了 **document indexing** 和 **document retrieval** 的一种方法。这种方法是对 **tfi-idf** 方法一种具体模型上的改进，整体看起来相对简单。
5. **A Translation Model for Matching Reviews to Objects**: 冲着 **Translation Model** 去看这篇文章的。文章中的翻译模型和 **uiuc** 中的翻译模型完全一样，通过这篇文章希望能够看懂 **Translation Model**。具体的翻译模型比较容易理解，对模型中参数的估计采用的 **EM** 算法。这篇文章对于 **uiuc** 的工作来说有很强的参考性，我刚看完翻译模型后面会全部看完的。

下周继续看 **information retrieval** 方面的论文，刚入门所以要看不少的论文~